الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic and Republic of Algeria

Ministry of Higher Education and Scientific Research
National School of Cyber Security
Department of Foundation Training

وزارة التعليم العالي والبحث العلمي
المدرسة الوطنية العليا في الأمن السيبراني
قسم التكوين القاعدي

First semester final exam

Level: 1st Year
Material: Algorithms and Static Data Structures

Date: 15/01/2025
Duration: 02h00

Context: This exam consists of a single comprehensive problem divided into many interconnected parts. It is strongly recommended to use modularity and to call modules across the different parts. You can use the predefined functions **pow(x, p)** to calculate x^p and **log2(n)** to calculate $\log_2(n)$.

Exercise 01: (06 points)

1. Write a module that rearranges the digits of a positive integer **n** to form the **largest possible number** and deal with this number as a **base password** (03 points).
2. Write a module to evaluate the **entropy** of the generated number by analyzing the count of unique digits it contains using the following formula. The entropy value indicates the security level of the password as follows: **Weak** if $Entropy \leq 2$, and **Strong** otherwise (02 points).

$$Entropy = \log_2(number\ of\ unique\ digits)$$

3. Write the main algorithm (01 point).

Exercise 2: (04 points)

1. Write a module to apply a **transformation** by reversing the digits of the **base password** (generated previously) and then **subtracting** the **smallest possible number** (obtained by rearranging the digits of **n** in the smallest number) from the **base password** (02 points).
2. Write the main algorithm (02 points).

Example

- **Input**: $n = 317$
- **Smallest number**: 137
- **Base password**: 731
- **Subtract**: $731 - 137 = 594$ (This is the final **obfuscated password)**

Exercise 3 (05 points)

1. Write a module that calculates the sum of the digits raised to the power of their index (index starts from 1). This value will serve as an **encryption key** for further transformations (02 points).
2. After that, shuffle the digits of **n** based on the encryption key (by rotating the digits left by a number of positions equal to the last digit of the encryption key) (02 points).
3. Write the main algorithm (01 points).

Example

$n = 442775$

$$Encryption\ key = 4^1 + 4^2 + 2^3 + 7^4 + 7^5 + 5^6$$
$$= 4 + 16 + 8 + 2401 + 16807 + 15625 = 19901$$

Use the last digit (i.e., 1) of the *Encryption key* to rotate the digits of **n** by 1 position
Shuffled number= 427754

Exercise 04: (05 points)

1. Write a module that transforms a numeric password (positive Integer **n**) into a more secure representation by analyzing **consecutive repetitions** of its digits (04 points).
2. Write the main algorithm (01 points).

Example

- $n = 44455522$
- $Output = 435322$

Good Luck